REMARKS

This response is submitted in response to an Office Action transmitted on June 20, 2007. Claims 1-38 were pending at the time the Office Action was issued. Applicants hereby amend claims 1, 7, 12-13, 15-19, 28-29, 31, and 34. Claims 1-38 remain pending.

In the interest of reducing the issues to be considered in this response, the following remarks focus principally on the patentability of independent claims 1, 19, and 28-29. The patentability of each of the dependent claims is not necessarily separately addressed in detail. However, Applicants' decision not to discuss the differences between the cited art and each dependent claim should not be considered as an admission that Applicants concur with the conclusions set forth in the Office Action that these dependent claims are not patentable over the disclosure in the cited references. Similarly, Applicants' decision not to discuss differences between the prior art and every claim element, or every comment set forth in the Office Action, should not be considered as an admission that Applicants concur with the interpretation and assertions presented in the Office Action regarding those claims. Indeed, Applicants believe that all of the dependent claims patentably distinguish over the references cited. Moreover, a specific traverse of the rejection of each dependent claim is not required, since dependent claims are patentable for at least the same reasons as the independent claims from which the dependent claims ultimately depend.

I. EXAMINER INTERVIEW

Applicants respectfully express their appreciation to Examiner Cao for the telephone interview held on August 23, 2007, during which the Examiner discussed the disposition of this case with the undersigned attorney. Specifically, the Examiner and the undersigned attorney discussed the creation of "parameter definitions" using "class declarations," such as class declarations associated with object-oriented programming. The Examiner and the undersigned attorney also discussed the lack of disclosures regarding "class declarations" in U.S. Patent No. 6,286,035 to Gillis et al. (hereinafter "Gillis"). However, agreement was not reached regarding the allowability of the claims.

II. PROVISIONAL DOUBLE-PATENTING REJECTIONS

Claims 1-38 are provisionally rejected on the ground of non-statutory obviousness-type double patenting as being unpatentable over claims 1-8, 10-27, and 29-37 of co-pending Application No. 10/883,373. Applicants submit herewith a terminal disclaimer to overcome these rejections. Accordingly, Applicants respectfully request reconsideration and withdrawal of these rejections.

III. REJECTIONS UNDER 35 U.S.C. § 101

Claims 1-18, 29-36, and 38 are rejected under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. Applicants have amended each of the claims 1-18, 29-36, and 38 to recite a "computer readable storage medium" instead of a "computer readable medium." Accordingly, Applicants respectfully request reconsideration and withdrawal of the objections to these claims.

IV. REJECTIONS UNDER 35 U.S.C. § 102

Claims 1-2, 4-7, 10, 13-17, 19-23, 26, and 27-38 were rejected under 35 U.S.C. § 102(b) as having been anticipated by Gillis. Respectfully, Applicants traverse the rejections, and submit that the claims are allowable over the reference cited to Gillis for at least the reasons explained in detail below.

Claim 1-2, 4-7, 10, and 13-17

Claims 2, 4-7, 10, and 13-17 depend from Claim 1. Claim 1, as amended, recites:

> 1. A computer readable medium encoded with a data structure, comprising:
>
> a parameter definition for at least one input parameter, the parameter definition being configured to enable identification of an appropriate input for the at least one input parameter, *wherein the parameter definition is created via a class declaration within the data structure*; and
>
> an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising at least one live object,
>
> wherein the instruction-based mechanism is further configured to process the at least one input parameter based on the identified appropriate input when the data structure becomes instantiated into an object. (emphasis added).

First, Applicants respectfully submit that amended claim 1 is fully supported under § 112, 1st paragraph, at least by (1) page 27, line 4 through page 28, line 12; (2) page 50, lines 5-10; (3) page 51, lines 20-25; (4) page 53, line 14 through page 54, line 23; and (5) page 74, lines 4-16 of the Applicants' specification, and by Figures 7 and 15-16 of the Applicants' drawings.

Second, Applicants respectfully submit that Gillis fails to teach or suggest the computer readable medium recited in claim 1. Specifically, Gillis fails to teach or suggest, "a parameter definition for at least one input parameter...*wherein the parameter definition is created via a class declaration within the data structure*," as recited in claim 1. (emphasis added).

Instead, Gillis discloses a software application that stores command parameters information in data tables. (Column 5, Lines 40-44). The data tables are either stored in a "message database 50", such as in a Microsoft ACCESS database, or as "plain text." (Column 4, Lines 20-25; Column 5, Lines 14-16). Additionally, Gillis discloses that the command parameter data table, "Tparam," may be used by a message parsing engine 100 to validate received command messages. (Column 7, Lines 10-13). However, Gillis does not disclose the use of "class declaration", as recited in claim 1, with the command parameters stored in its data tables.

Accordingly, for at least the reasons stated above, Gillis fails to anticipate claim 1. Moreover, since claims 2, 4-7, 10, and 13-17 depend from claim 1, they are at least allowable due to their dependency, as well as due to additional limitations recited.

Specifically, claim 7 is further allowable over Gillis. Claim 7, as amended, recites:

> 7. The computer readable medium of claim 6, wherein the parameter definition comprises a data type and a name for the expected input parameter, and wherein the mechanism further coerces the value having a first data type into a converted value having a second data type specified in the definition.

Applicants respectfully submit that amended claim 7 is fully supported under § 112, 1st paragraph, at least by page 53, lines 2-5. Additionally, Gillis does not teach or suggest a computer readable medium, as recited in amended claim 7, "wherein the mechanism further coerces the value having a first data type into a converted value *having a second data type specified in the definition.*" (emphasis added). Instead, Gillis discloses the translation of a value into "a value to be placed in the data structure" of a message. (Column 8, Lines 36-39). In other words, Gillis teaches the conversion of a value into another value, but does not teach the conversion of the data type of the value into another data type. Accordingly, claim 7 is further allowable over Gillis.

Moreover, claim 14 is also further allowable over Gillis. Claim 14 recites:

> 14. The computer readable medium of claim 1, wherein the mechanism comprises a method inherited from a class provided within a runtime environment.

Gillis does not teach or suggest a computer readable medium, as recited in amended claim 14, "wherein the mechanism comprises *a method inherited from a class* provided within a runtime environment." (emphasis added). Instead, Gillis discloses a "do Validate ()" function that parses a message, validates each part of the message, and transforms the message into a data structure for processing by the system code 90. (Column 4, Lines 57-60). However, since Gillis does not teach or suggest the declaration of classes, Gillis cannot teach or suggest that the "do Validate ()" function is "inherited from a class," as recited in claim 14. Accordingly, claim 14 is further allowable over Gillis.

Claims 19-23 and 26

Claims 20-23 and 26 depend from claim 19. Claim 19, as amended, recites:

19. A computer-executable method for populating parameters declared within a data structure, the method comprising:

obtaining an expected name for a parameter, the expected name being assigned *in a class declaration* for the parameter within a data structure;

identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;

retrieving a value associated with the label; and

assigning the value to the parameter. (emphasis added).

First, Applicants respectfully submit that amended claim 19 is fully supported under § 112, 1st paragraph, at least by (1) page 27, line 4 through page 28, line 12; (2) page 50, lines 5-10; (3) page 51, lines 20-25; (4) page 53, line 14 through page 54, line 23; and (5) page 74, lines 4-16 of the Applicants' specification, and by Figures 7 and 15-16 of the Applicants' drawings.

Second, Applicants respectfully submit that Gillis fails to teach or suggest the computer-executable method recited in claim 19. Specifically, Gillis does not teach or suggest, "obtaining an expected name for a parameter, the expected name being assigned *in a class declaration* for the parameter within a data structure," as recited in claim 19. (emphasis added).

Instead, Gillis discloses associating a unique key (*e.g.*, database array index) to a parameter. (Column 7, Lines 44-48). Gillis also discloses finding a description of the parameter by comparing the unique key of a received parameter with the data found in the tables of the database. (Column 7, Lines 60-63). Nevertheless, Gillis does not teach or suggest assigning the unique key to the parameter by "class declaration," as recited in claim 19.

Accordingly, for at least the reason stated above, Gillis fails to anticipate claim 19. Moreover, since claims 20-23 and 26 depend from claim 19, they are at

least allowable due to their dependency, as well as due to additional limitations recited.

Claim 28

Claim 28, as amended, recites:

> 28. A system the handles input parameters, the system comprising:
> a means for processing; and
> a memory means, the memory means being allocated for a plurality of computer-executable instructions which are loaded into the memory means for execution by the means for processing, the computer-executable instructions performing a method comprising:
> a means for obtaining an expected name for a parameter, the expected name being assigned *in a class declaration* for the parameter within a data structure;
> a means for identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;
> a means for retrieving a value associated with the label; and
> a means for assigning the value to the parameter. (emphasis added).

First, Applicants respectfully submit that amended claim 28 is fully supported under § 112, 1st paragraph, at least by (1) page 27, line 4 through page 28, line 12; (2) page 50, lines 5-10; (3) page 51, lines 20-25; (4) page 53, line 14 through page 54, line 23; and (5) page 74, lines 4-16 of the Applicants' specification, and by Figures 7 and 15-16 of the Applicants' drawings.

Second, Applicants incorporate the argument presented above in response to the rejection of 19 by analogy, and submit that Gillis fails to teach or suggest the system recited in claim 28. Specifically, Gillis does not teach or suggest, "a means for obtaining an expected name for a parameter, the expected name being

assigned *in a class declaration* for the parameter within a data structure," as recited in claim 19. (emphasis added).

Instead, Gillis discloses associating a unique key (*e.g.,* database array index) to a parameter. (Column 7, Lines 44-48). Gillis also discloses finding a description of the parameter by comparing the unique key of a received parameter with the data found in the tables of the database. (Column 7, Lines 60-63). However, Gillis does not teach or suggest assigning its unique key to the parameter by "class declaration." Accordingly, for at least the reason stated above, Gillis fails to anticipate claim 28.

Claims 29-38

Claims 30-38 depend from claim 29. Claim 29, as amended, recites:

> 29. A computer readable medium encoded with a data structure that provides a template for creating an application, the data structure comprising:
> a name identifying an application *that is included in a declared parent class* provided by an object-based environment;
> at least one member configured to receive one or more sets of input, wherein each set of input comprises at least one live object; and
> a method associated with the one or more sets of input,
> wherein the declared parent class is configured to provide processing that executes the method for each set of input received for the at least one member when the name of the application is invoked. (emphasis added).

First, Applicants respectfully submit that amended claim 29 is fully supported under § 112, 1st paragraph, at least by (1) page 27, line 4 through page 28, line 12; (2) page 50, lines 5-10; (3) page 51, lines 20-25; (4) page 53, line 14 through page 54, line 23; and (5) page 74, lines 4-16 of the Applicants' specification, and by Figures 7 and 15-16 of the Applicants' drawings.

Second, Applicants respectfully submit that Gillis does not teach or suggest the computer readable medium of claim 29. Applicants first submit that Gillis does not teach or suggest, "a name identifying an application that is included in a declared parent class provided by an object-based environment," as recited in claim 29. (emphasis added).

Third, Gillis discloses a parsing engine object code 100 that is downloadable into a target computer 30. (Column 7, Lines 36-37). However, Gillis does not teach or suggest that the parsing engine object code 100 includes a "declared parent class."

Second, since Gillis does not disclose that the parsing engine object code 100 includes a "declared parent class," Gillis also does not teach or suggest, "wherein the declared parent class is configured to *provide processing that executes the method for each set of input received for the at least one member when the name of the application is invoked*," as recited in claim 29.

Therefore, for at least the reasons stated above, Gillis fails to anticipate claim 29. Moreover, since claims 30-38 depend from claim 29, they are at least allowable due to their dependency, as well as due to additional limitations recited.

Specifically, claim 34 is further allowable over Gillis. Claim 34 recites:

> 34. The data structure of claim 29, wherein the declared parent class further provides a mapping process that allows a specified alias for the identifier.

First, Applicants respectfully submit that amended claim 28 is fully supported under § 112, 1st paragraph, at least by page 52, lines 7-18. Second, Gillis does not teach or suggest a data structure, as recited in amended claim 34, "wherein the declared *parent class* further provides a mapping process that allows a specified alias for the identifier." (emphasis added). Instead, Gillis discloses

associating a unique key (*e.g.*, database array index) to a parameter. (Column 7, Lines 44-48). However, Gillis does not teach or suggest that this association is provided by a "declared parent class." Accordingly, claim 34 is further allowable over Gillis.

V. REJECTION UNDER 35 U.S.C. § 103 OF CLAIMS 3, 8, 12, 25, AND 27

Claims 3, 8, 12, 25, and 27 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Gillis in view of the U.S. Patent 6,405,365 to Lee *et al.* (hereinafter "Lee"). Applicants respectfully traverse the rejections.

Claims 3, 8, and 12

Claims 3, 8, and 12 depend from Claim 1. Claim 1, as amended, recites:

> 1. A computer readable medium encoded with a data structure, comprising:
>
> a parameter definition for at least one input parameter, the parameter definition being configured to enable identification of an appropriate input for the at least one input parameter, *wherein the parameter definition is created via a class declaration within the data structure*; and
>
> an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising at least one live object,
>
> wherein the instruction-based mechanism is further configured to process the at least one input parameter based on the identified appropriate input when the data structure becomes instantiated into an object. (emphasis added).

Applicants respectfully incorporate the argument present above in response to the rejection of claim 1 under 35 U.S.C. § 102(b) by analogy. Accordingly, applicants submit that Gillis does not teach or disclose, "a parameter definition for

at least one input parameter...*wherein the parameter definition is created via a class declaration within the data structure*," as recited in claim 1. (emphasis added).

Moreover, the deficiencies of Gillis are not remedied by Lee. Instead, Lee discloses an Instruction File 150, and generating a Command-Field Record for an instruction "by extracting from the instruction the Command-Field Values specified by the first Syntax Record." (Figure 2; Column 8, Lines 15-20).

Accordingly, the cited references to Gillis and Lee, whether individually or in combination, do not teach or suggest the computer readable medium recited in claim 1. Further, since claims 3, 8, and 12 depend from claim 1, they are also allowable over the cited references at least due to their dependency, as well as due to additional limitations recited.

Claims 25 and 27

Claims 25 and 27 depend from Claim 19. Claim 19, as amended, recites:

> 19. A computer-executable method for populating parameters declared within a data structure, the method comprising:
> obtaining an expected name for a parameter, the expected name being assigned *in a class declaration* for the parameter within a data structure;
> identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;
> retrieving a value associated with the label; and
> assigning the value to the parameter. (emphasis added).

Applicants respectfully incorporate the argument present above in response to the rejection of claim 19 under 35 U.S.C. § 102(b) by analogy. Accordingly, applicants submit that Gillis does not teach or disclose, "obtaining an expected

name for a parameter, the expected name being assigned *in a class declaration* for the parameter within a data structure," as recited in claim 19. (emphasis added).

Moreover, the deficiencies of Gillis are not remedied by Lee. Instead, Lee discloses an Instruction File 150. (Figure 2; Column 8, Lines 15-20). Accordingly, the cited references to Gillis and Lee, whether individually or in combination, do not disclose, teach or fairly suggest the computer-executable method recited in claim 19. Further, since claims 25 and 27 depend from claim 19, they are also allowable over the cited references at least due to their dependency, as well as due to additional limitations recited.

VI. REJECTION UNDER 35 U.S.C. § 103 OF CLAIM 9

Claim 9 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Gillis in view of Lee, and in further view of the Jones (Parse and Validate Command Line Parameters with VB.NET) (hereinafter "Jones"). Applicants respectfully traverse the rejection.

Claim 9 depends from claim 1. Claim 1, as amended, recites:

> 1. A computer readable medium encoded with a data structure, comprising:
>
> a parameter definition for at least one input parameter, the parameter definition being configured to enable identification of an appropriate input for the at least one input parameter, *wherein the parameter definition is created via a class declaration within the data structure*; and
>
> an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising at least one live object,
>
> wherein the instruction-based mechanism is further configured to process the at least one input parameter based on the identified appropriate input when the data

> structure becomes instantiated into an object. (emphasis added).

Applicants respectfully incorporate the argument present above in response to the rejection of claim 8 under 35 U.S.C. § 103(a) by analogy. Accordingly, applicants submit that the cited references to Gillis and Lee, whether individually or in combination, do not disclose, teach or fairly suggest, "a parameter definition for at least one input parameter...*wherein the parameter definition is created via a class declaration within the data structure*," as recited in claim 1. (emphasis added).

Moreover, the deficiencies of Gillis are not remedied by Jones. Instead, Jones discloses that in VB.Net, a passed command line may be obtained via the Command () function. (Page 2, Paragraph 2, Lines 1-2). However, Jones does not teach the creation of a parameter definition to a parameter via a class declaration. Accordingly, the cited references (Gillis, Lee and Jones), whether individually or in combination, do not disclose, teach or fairly suggest the computer readable medium recited in claim 1. Further, since claim 9 depends from claim 1, it is also allowable over the cited references at least due to its dependency, as well as due to additional limitations recited.

VII. REJECTION UNDER 35 U.S.C. § 103 OF CLAIMS 11 AND 18

Claims 11 and 18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Gillis in view of Lee, and in further view of U.S. Patent No. 6,658,625 to Allen et al. (hereinafter "Allen"). Applicants respectfully traverse the rejections.

Claims 11 and 18 depend from Claim 1. Claim 1, as amended, recites:

1. A computer readable medium encoded with a data structure, comprising:

a parameter definition for at least one input parameter, the parameter definition being configured to enable identification of an appropriate input for the at least one input parameter, *wherein the parameter definition is created via a class declaration within the data structure*; and

an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising at least one live object,

wherein the instruction-based mechanism is further configured to process the at least one input parameter based on the identified appropriate input when the data structure becomes instantiated into an object. (emphasis added).

Applicants respectfully incorporate the argument present above in response to the rejection of claim 1 under 35 U.S.C. § 103(a) by analogy. Accordingly, applicants submit that the cited references to Gillis and Lee, whether individually or in combination, do not disclose, teach or fairly suggest, "a parameter definition for at least one input parameter...*wherein the parameter definition is created via a class declaration within the data structure*," as recited in claim 1. (emphasis added).

Moreover, the deficiencies of Gillis are not remedied by Allen. Instead, Allen discloses a parser that is capable of validate and parse an XML document using the Document Type Definition (DTD) of the document. (Column 19, Lines 35-36). Allen also discloses a Program Call Markup Language (PCML) data description that provides configurable data definitions. (Column 6, Lines 35-42). However, this disclosure also does not remedy the deficiencies of Gillis as it is not related to the use of class declarations.

Accordingly, the cited references (Gillis, Lee and Allen), whether individually or in combination, do not disclose, teach or fairly suggest the

computer readable medium recited in claim 1. Further, since claims 11 and 18 depend from claim 1, they are also allowable over the cited references at least due to their dependency, as well as due to additional limitations recited.

VIII. REJECTION UNDER 35 U.S.C. § 103 OF CLAIM 24

Claim 24 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Gillis in view of Allen. Applicants respectfully traverse the rejection.

Claim 24 depends from claim 19. Claim 19, as amended, recites:

> 19. A computer-executable method for populating parameters declared within a data structure, the method comprising:
>
> obtaining an expected name for a parameter, the expected name being assigned in a class declaration for the parameter within a data structure;
>
> identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;
>
> retrieving a value associated with the label; and
>
> assigning the value to the parameter.

Applicants respectfully incorporate the argument present above in response to the rejection of claim 19 under 35 U.S.C. § 102(b) by analogy. Accordingly, applicants submit that Gillis does not teach or disclose, "obtaining an expected name for a parameter, the expected name being assigned *in a class declaration* for the parameter within a data structure," as recited in claim 19. (emphasis added).

Moreover, the deficiencies of Gillis are not remedied by Allen. Instead, Allen discloses a parser that is capable of validate and parse an XML document using the Document Type Definition (DTD) of the document. (Column 19, Lines 35-36). However, this disclosure of Allen does not teach the use of class declarations.

Accordingly, the cited references to Gillis and Allen, whether individually or in combination, do not disclose, teach or fairly suggest the computer readable medium recited in claim 19. Further, since claim 24 depends from claim 19, it is allowable over the cited references at least due to its dependency, as well as due to additional limitations recited.

CONCLUSION

For the foregoing reasons, Applicants respectfully submit that claims 1-38 are now in condition for allowance. If there are any remaining matters that may be handled by telephone conference, the Examiner is kindly invited to contact the undersigned attorney at the telephone number listed below.

Respectfully Submitted,

Dated: 10-4-07 By: 

Elliott Y. Chen
Reg. No. 58,293
Lee & Hayes, PLLC
421 W. Riverside Ave, Suite 500
Spokane, WA 99201
Phone: (206) 315-4001 x104
or (206) 315-7914
Fax: (206) 315-4004

Enclosure: Terminal Disclaimer